Redundant Via Insertion under Timing Constraints

Chi-Wen Pan, Yu-Min Lee
National Chiao Tung University, Hsinchu, Taiwan
wayne1234.cm97g@nctu.edu.tw, yumin@cm.nctu.edu.tw

Abstract—Redundant via insertion is a useful technique to alleviate the yield loss and elevate the reliability of VLSI designs. While extra vias are inserted into the design, the electronic properties of designed circuit might be altered, and the circuit timing might be changed and needs to be efficiently re-analyzed. Therefore, a fast timing (incremental timing) analyzer is required to assist the redundant via insertion procedure.

This work develops an efficient redundant via insertion method under timing constraints. Firstly, an effectively incremental circuit timing analysis method is developed, and the redundant via insertion task is transformed into a mixed bipartite-conflict graph matching problem. Then, the insertion problem is solved by a timing-driven minimum weighted matching algorithm.

The experimental results show that the developed algorithm can achieve 3.2% extra insertion rates over the method without considering timing effects, which all redundant vias would be removed if the timing of that net does not meet the timing requirements, in average. In addition, the developed incremental timing analysis mechanism can speed up the runtime of redundant via insertion procedure under timing constraints by over 34 times in average.

I. INTRODUCTION

As the feature size continuously scales down, yield becomes an important issue for the current modern design. Yield-loss comes from many physical factors, and via failure is one major factor caused by electromigration, thermal stress and random defects. Partial via failure increases circuit resistances that result in the unexpected timing delay, and complete via failure can break the net connection and lead to the inaccurate signal that makes the circuit fail. Without violating any design rules, inserting a redundant via (RV) adjacent to a single via as a safeguard is a widely recommended method for improving the circuit yield and reliability [1], [2]. With the double vias, the via failure rate can be dramatically reduced as reported in [3].

Many researches performed the RV insertion procedure during the routing stage [4]–[6]. Xu et al. [4] developed a maze routing with considering RV insertion for the yield enhancement. Yao et al. [5] minimized the via usages and inserted redundant vias in the routing stage to improve the multilevel routing framework. Chen et al. [6] proposed a full-chip gridless routing method with considering double-via insertion based on the bipartite matching graph algorithm. However, they stacked the vertical vias as one stack via that the solution space was reduced. Besides, many researches worked on the post-routing stage [7], [8]. Lee et al. [7] proposed a zero-one integer linear program to solve double-cut via insertion problem and handle the via density constraint. Lei et al. [8] transformed the RV insertion task into a mixed bipartite matching graph and presented a heuristic minimum weighted matching algorithm to solve the problem. They also developed the wire spreading method to insert redundant vias for dead vias.

Inserting RV into the circuit might vary the timing characteristic of design. Luo et al. [9] created a set of untouched nets such as the timing critical nets or the nets specified by customers etc., and utilized the geotopological technology to insert redundant vias. Although they considered the timing issue, the insertion algorithm cannot guarantee whether the circuit timing is degraded or not after the insertion. Chiang et al. [10] developed a two-phase insertion approach method with considering timing constraints. However, they only checked the timing behavior after executing the insertion procedure and simply got rid of inserted redundant vias violating timing constraints. In other words, they didn’t immediately update the timing behavior during the RV insertion procedure. Therefore, the insertion rate is degraded.

Recently, Lin et al. [11] pointed out that “How to tackle the timing issue more accurately during double-via insertion is still worthy further study.”. The experimental results illustrated in TABLE IV also show that the ratio range of net sink node delay differences after performing the conventional (without considering the timing effects) redundant via insertion method [8] on test circuits can be 35% ~ 58%. Therefore, it is necessary to develop timing-driven redundant via insertion methods.

Because the timing information should be frequently predicted and updated during the RV insertion process, a fast timing analysis method is required. In this work, firstly, an incremental analysis method is developed to effectively perform the timing analysis and predict the timing behavior for the RV insertion process. The developed incremental analysis method does not need to recalculate the entire net timing to predict the timing effect of adding an extra via. It only needs to analyze the timing influence induced by the modified circuit parts. Therefore, the computational load of timing analysis can be dramatically reduced. After that, the proposed incremental timing analysis method is incorporated with the mixed bipartite-conflict (MBC) graph [8] for developing a timing-driven minimum weighted matching (t-MWM) algorithm to solve the RV insertion task.

To the best of our knowledge, this is the first redundant via insertion method that truly considers the timing effects of inserted vias during the insertion procedure.

The paper is organized as follows. The redundant via insertion problem under timing constraints is formulated in section II. Then, an efficient incremental timing analysis method for the redundant via insertion problem is detailed in section III, and the MBC graph [8] is briefly presented in section IV. After that, the developed t-MWM algorithm and the experimental results are presented and discussed in section V and section VI, respectively. Finally, the conclusion is given in section VII.
II. PROBLEM FORMULATION

Given a post-routing design, its routed netlist and each net timing constraint, the RV insertion problem under timing constraints is to simultaneously insert extra vias as many as possible to the design and satisfy the given timing constraints. The problem can be formulated as follows.

$$\max \sum_{i}^{N} M_i \sum_{k}^{P_i} RV_{ik},$$

subject to

$$D_j^i \leq D_j^i, \quad \forall (\text{net } i) \ \& \ (\text{sink } j \text{ of net } i)$$

Here,

$$RV_{ik} = \begin{cases} 1 & \text{if an RV is inserted to a single via } k \text{ of net } i \\ 0 & \text{otherwise.} \end{cases}$$

$N$ is the number of nets in the given design, $M_i$ is the number of single vias on net $i$, $D_j^i$ is the propagation delay from the source of net $i$ to its sink $j$, and $D_j^i$ is the timing constraint of sink $j$ on net $i$.

III. INCREMENTAL TIMING ANALYSIS

A. Incremental Timing Formula

A simple RC tree shown in Fig. 1 is used to describe the closed form of the incremental timing formula. Given an original RC tree shown in Fig. 1.(a) and its locally modified RC tree shown in Fig. 1.(b), their Elmore delay values from node 0 to node $n$ can be calculated as

$$t_{n}^{(a)} = R_1 \sum_{i=1}^{6} C_i + R_2 \sum_{j=2}^{6} C_j + R_3 \sum_{k=3}^{4} C_k + R_4 C_4$$

$$t_{n}^{(b)} = t_{n}^{(a)} + \Delta t_n.$$

Here, their timing difference $\Delta t_n$ can be derived as

$$\Delta t_n = \Delta C_i \sum_{i=1}^{2} R_i + \Delta C_3 \sum_{j=1}^{3} R_j + \Delta C_5 \sum_{k=1}^{2} R_k$$

$$+ \Delta R_2 \sum_{i=2}^{6} C_i + \Delta R_3 \sum_{j=3}^{4} C_j$$

$$+ \Delta R_2 \sum_{i=2,3,5} C_i + \Delta R_3 \Delta C_3.$$

Here, each $\Delta C_i$ is the difference value of nodal capacitance at node $i$ after and before via insertion, and each $\Delta R_j$ is the difference value of branch resistance on branch $j$ after and before via insertion. The expression of (1) consists of three types. Each term in the first line is the product of a delta nodal capacitance and its original upstream common path equivalent resistance. The original upstream common path equivalent resistance is the sum of original overlapped segment resistances between the paths from source node 0 to the nodal-capacitance changed node and from source node 0 to node $n$ in Fig. 1.(a). Each term in the second line is the product of a delta branch-resistance (on the path from source node 0 to node $n$) and its original downstream equivalent capacitance. Each term in the last line is an interactive term that is the product of a delta branch-resistance (on the path from source node 0 to node $n$) and its delta downstream equivalent capacitance.

With (1), the timing difference $\Delta t_n$ for each node $n$ after a RC tree is locally modified can be generally formulated as follows.

$$\Delta t_n = \sum_{i} R_{ni} \Delta C_i + \sum_{j \in P_n} \Delta R_j C_{j^*} + \sum_{j \in P_n} \Delta R_j \Delta C_{j^*}$$

Here, $R_{ni}$ is the original upstream common path equivalent resistance between $P_n$ and $P_i$, and $P_n/P_i$ is the routed path from source node to node $n/i$. $C_{j^*}$ is the original equivalent downstream capacitance seen from branch $j$, and $\Delta C_{j^*}$ is the difference value of equivalent downstream capacitance seen from branch $j$ after and before via insertion.

By utilizing the timing difference formula of node $n$ shown in (2), it can be very efficient to predict each specific net sink node delay without recalculating all node delays of a specific net.

B. Redundant Via Shapes

According to the connection between wires and the single via, the structures of redundant vias can be categorized into two different shapes. If a single via only connects two wire segments, it is called an L-shape since this structure looks like an alphabet letter ‘L’. If a single via connects more than two wire segments, it is called a T-shape since it looks like an alphabet letter ‘T’. Moreover, according to the characteristic of redundant vias, they can be divided into two types, on-track RV and off-track RV. The on-track RV is placed on the original net and needs only one extra wire segment, and an off-track RV is placed out of the original net and needs to add two extra wire segments.

Since L-shape RVs and T-shape RVs are cyclic circuits, in order to effectively calculate their Elmore delay values, several fundamental circuit transformation techniques are utilized to make them acyclic. The equivalent acyclic RC circuits of each L-shape RV and each T-shape RV are derived as follows to incrementally update the circuit timing efficiently.

B.1. L-shape

The L-shape RVs contain three templates that two types are on-track RVs, and one type is off-track RV. These templates and their corresponding RC trees are shown in Fig. 2. Fig. 3 shows the equivalent acyclic RC circuits of above three L-shape RV templates, and their related $\Delta R$ values and $\Delta C$ values are summarized in TABLE I.

B.2. T-shape

According to the transformation methods, T-shape RVs can be divided into simple T-shape RVs and complicated T-shape


RVs precisely. The simple T-shape RVs contain three templates that two are on-track RVs, and one is off-track RV. Utilizing the similar circuit transformation techniques as L-shape RV templates, they can be transformed to corresponding equivalent RC tree structures. The simple T-shape RV templates and their equivalent RC trees are shown in Fig. 4. Fig. 5 represents equivalent acyclic RC circuits of simple T-shape RV templates. Although the equivalent circuit structure of simple T-shape RV is different with that of L-shape RV, they have the same delta RC values as shown in Fig. 1 because the extra wire segment in each simple T-shape RV has no impact on the equivalent circuit transformation.

The delta-wye transformation is needed for transforming the rest T-shape RV templates to be acyclic circuits with extra resistance and capacitance parameters. Their equivalent RC tree structures are more complicated than those of simple T-shape RV templates. We call them complicated T-shape RVs. The complicated T-shape RV templates and their equivalent RC trees are shown in Fig. 6. Fig. 7 is the equivalent acyclic RC circuit of complicated T-shape RV templates, and their related ΔR values and ΔC values are summarized in TABLE. II.

### C. Fast Sink Node Timing Check

To efficiently check whether the timing of each related sink node will violate its timing constraint or not if a related RV is inserted, the timing difference formula Δ$t_{in}$ shown in (2) and the related ΔR values and ΔC values for L-shape RVs and T-shape RVs presented in TABLE. I and TABLE. II, respectively, are integrated to build a fast sink node timing check algorithm shown in Fig. 9. By executing the algorithm shown in Fig. 9, the timing of sink node as if adding an RV candidate can be fast checked.

**Remark:** The developed fast sink node timing check algorithm can be easily extended to perform the path delay check by using the static timing analysis method to calculate each gate’s require arrival time (RT), arrival time (AT), and slack. By setting each gate’s slack to be non-negative as a timing constraint during the RVI algorithm, one inserted redundant via might change the gate’s RT or AT on the net that can be verified by the fast sink node timing check method. After that, the
TABLE II
\( \Delta R \) VALUES AND \( \Delta C \) VALUES OF THE EQUIVALENT ACYCLIC RC CIRCUIT OF COMPLICATED T-SHAPE RV.

<table>
<thead>
<tr>
<th>Complicated T-Shape RV</th>
<th>( \Delta R_1 )</th>
<th>( \Delta R_2 )</th>
<th>( \Delta R_3 )</th>
<th>( \Delta R_4 )</th>
<th>( \Delta R_5 )</th>
<th>( \Delta R_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>on-track type I</td>
<td>(-r_{u1x})</td>
<td>(+r_{u1x})</td>
<td>(+r_{u1x})</td>
<td>(-r_{u1x})</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>on-track type II</td>
<td>0</td>
<td>(+r_{u1x})</td>
<td>(+r_{u1x})</td>
<td>(-r_{u1x})</td>
<td>-r_{u3x}</td>
<td>0</td>
</tr>
<tr>
<td>on-track type III</td>
<td>0</td>
<td>(+r_{u1x})</td>
<td>(+r_{u1x})</td>
<td>(-r_{u1x})</td>
<td>0</td>
<td>-r_{u2x}</td>
</tr>
</tbody>
</table>

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{Complicated T-Shape RV} & \Delta C_1 & \Delta C_2 & \Delta C_3 & \Delta C_4 & \Delta C_5 & \Delta C_6 \\
\hline
\text{on-track type I} & \(-v_{u1x}\) & \(-w_{u1x}\) & \(-w_{u1x}\) & 0 & 0 & 0 \\
\text{on-track type II} & 0 & \(-w_{u1x}\) & \(-w_{u1x}\) & 0 & 0 & 0 \\
\text{on-track type III} & 0 & \(-w_{u1x}\) & \(-w_{u1x}\) & 0 & 0 & -\frac{v_{u2x}}{2} \\
\hline
\end{array}
\]

Fig. 6. Complicated T-shape RV. (a) on-track type I. (b) on-track type II. (c) on-track type III.

Fig. 7. The equivalent acyclic RC circuit of complicated T-shape RV.

delay effect is passed to the upstream and downstream path, and the circuit delay is calculated and checked. With this method, we can effectively update the circuit timing, keep the slack being non-negative, and has no any timing violation.

D. Runtime Complexity Analysis

The incremental timing analysis method can analyze the timing difference in \( O(n_{\text{sink}}) \) time if a relation topology table between vias and sinks can be constructed in advance. Here, \( n_{\text{sink}} \) is the number of sink nodes on the net. Once the developed algorithm decides a redundant via to be inserted, the timing influence can be easily analyzed by using the look-up table. For example, after a redundant via has been inserted to via \( v_1 \) shown in Fig. 8, the timing influence on each sink can be analyzed by finding the common path between the via \( v_1 \) and each sink node.

\begin{align*}
  n_{cp}(v_1, S_1) &= v_1 \\
  n_{cp}(v_1, S_2) &= v_1 \\
  n_{cp}(v_1, S_3) &= n_1 \\
  n_{cp}(v_1, S_4) &= S_0 \\
  n_{cp}(v_1, S_5) &= S_0
\end{align*}

Here, \( n_{cp}(v, S) \) is the node that its upstream path is the common path of the via \( v \) and sink \( S \). With the formulas (2) and (3)–(7), the timing information can be efficiently predicted and updated. The conventional timing analysis method reanalyzes the timing values of all nodes on the net and needs to recalculate the downstream capacitance of each node. The runtime complexity of the conventional method is \( O(n) \), and \( n \) is the number of nodes on the net. Therefore, compared with the conventional method, the developed method is more efficiency.

IV. MBC GRAPH MATCHING PROBLEM

Lei et. al [8] proposed and defined the MBC graph matching problem with three essential definitions. The first definition constructs a via-candidate bipartite graph that includes the set of single vias on one side and the set of corresponding RV candidates on the other side. The second definition constructs a candidate relative graph that shows the relationship between RV candidates. The connection in graph means the conflict of two candidates because of design rules; in other words, they fight for one RV position. The third definition integrates two graphs
Algorithm: Sink Node Timing Check for Adding RV Candidate

Input:
- CRV: a redundant via candidate on net \( i \)
- \( S_1, \ldots, S_n \): sink nodes of net \( i \)
- \( t_{S_1}, \ldots, t_{S_n} \): original sink delays at \( S_1, \ldots, S_n \), respectively

Output:
- Satisfaction or Violation

01 Find the delta RC values of equivalent acyclic RC circuit of CRV by utilizing Fig. I or Fig. II
02 For sink nodes from \( S_1 \) to \( S_n \)
03 Calculate \( \Delta t_{S_j} \) by (2)
04 If \( t_{S_j} + \Delta t_{S_j} > \) the timing constraint of sink node \( S_j \)
05 Return Violation
06 End
07 Return Satisfaction

Fig. 9. Sink Node Timing Check for Adding RV Candidate Algorithm

V. TIMING DRIVEN REDUNDANT VIA INSERTION

Based on the MBC graph, the relationship between the single via set and RV candidate set are well connected. Here, we are going to present the developed timing-driven minimum weighted matching (t-MWM) algorithm to solve the MBC graph matching problem and use the timing check algorithm shown in Fig. 9 to efficiently check the sink node timing as if an RV candidate is inserted on a specific net.

A. Edge Weight Assignment for the MBC Graph

Given a constructed MBC graph, an edge weight value \( \omega(e) \) between a single via and its specific RV candidate needs to be assigned. This edge weight is determined by several properties between this single via and its relative RV candidates as follows.

\[
\omega(e) = \rho \cdot \left( \alpha \times F.N. + \beta \times C.D. + \gamma \times C.T. \right)
\]

Here, \( \alpha, \beta \) and \( \gamma \) are user-defined constants. The \( \rho \) is a timing violation indicator which is equal to infinity (i.e., this RV cannot be a candidate) if inserting the RV candidate will violate the timing-constraint, and it is equal to 1 if inserting the RV candidate won’t violate the timing-constraint. The key factors, \( F.N. \), \( C.D. \) and \( C.T. \) are

1) Feasible number (F.N.): It is the number of feasible RV candidates that a single via has, and the maximum number is 4.
2) Conflict degree (C.D.): It is the number of conflicts between one RV candidate and the rest RV candidates.
3) Candidate type (C.T.): For the manufacturing reason, we prefer to insert on-track RVs rather than off-track RVs. The value of \( C.T. \) is equal to 0 if it is an on-track RV candidate; otherwise, the value of \( C.T. \) is assigned to be 1.

B. Two Phase t-MWM Algorithm for a MBC Graph

The HMWM algorithm was used for solving the RV insertion problem in [8] but it is not suitable for the timing-driven redundant via insertion. To deal with timing constraints, we propose a two phase t-MWM algorithm that utilizes the properties of HMWM algorithm and considers timing issues. Fig. 11 shows the individual phase flowchart of the proposed two phase t-MWM algorithm. For each phase, similarly with HMWM algorithm [8], firstly, t-MWM algorithm assigns a suitable weight for each edge in the MBC graph and sorts the edge weights in the increasing order. Then, t-MWM algorithm picks up the RV candidate connected with the minimum edge weight, and the MBC graph is updated. Finally, above steps are repeated until there is no more RV candidate.

Compared with HMWM algorithm, the primary differences are that the edge weight assignment step considers the timing effect in t-MWM algorithm, and t-MWM algorithm re-analyzes the timing of modified circuit while performing the “edge weight update” step. Owing to inserting an RV will alter the net timing behavior, the timing effect on the same net should be recalculated. However, estimating the timing effect frequently is time-consuming and exhausted. To alleviate the computation load of updating timing effects, t-MWM algorithm utilizes the proposed timing check algorithm shown in Fig. 9 to check and update the sink-node timing for adding an RV candidate.

Phase 1 of t-MWM algorithm simplifies the problem and handles RV candidates that have no conflicts. In this phase, the edge weight is modified as (9) to accommodate this simple problem.

\[
\omega(e) = \max \left\{ \frac{t_{S_j} + \Delta t_{S_j}}{t_{S_j}}, \infty \right\}, \quad \text{if } \Delta t_{S_j} < 0, \forall \text{ sink node } j \text{ of the related net, } \text{(9)}
\]

Here, \( t_{S_j} \) is the original sink delay at sink node \( j \) of the related net, and \( \Delta t_{S_j} \) is the timing difference at sink node \( j \) of the related net after inserting a related RV.
Phase 1 is a pre-stage to increase the timing budget for other candidates in Phase 2. Without any conflicts of RV candidates, the candidates inserted in Phase 1 will not decrease the RV candidate number in Phase 2. Furthermore, according to (9), each picked RV can increase the timing-budget.

Phase 2 deals with the rest RV candidates excluding those inserted in Phase 1 and utilizes (8) for the edge weight assignment. The rest steps in Phase 2 are the same as those of Phase 1 shown in Fig. 11.

Remark: The concept of pre-increasing the timing budget in Phase 1 has a significant influence on the insertion number of Phase 2. For example, the single via $V_7$ in Fig. 10 has only one RV candidate $R_{T7}$ that conflicts to $R_{L6}$. The better choice is to pick up $R_{T6}$ and $R_{T7}$ to match single vias $V_6$ and $V_7$ individually. However, if we execute Phase 2 directly instead of performing Phase 1 firstly, the $\omega(\epsilon)$ related to $R_{T7}$ will be infinity if the timing constraint is violated after $R_{T7}$ is chosen to be inserted. In other words, $R_{T7}$ cannot be a candidate under timing constraints unless other inserted RVs at the same net can provide enough timing budget for $R_{T7}$.

As a result, if Phase 1 can enhance the timing budget, it is helpful for improving the insertion rate in Phase 2.

VI. EXPERIMENTAL RESULTS

The developed RV insertion method under timing constraints has been implemented in C++ programming language and tested on an Intel Xeon 5160 quad core 3.0-GHz Linux based machine with 32GB memory. The test designs are the routed MCNC benchmarks that were built by the authors of [6] with 0.18$\mu$m technology. The RC parasitic parameters are listed in TABLE III, and the resistance of via is equal to 6.4$\Omega$. The first two cases contain 4 metal layers and others have 3 metal layers. For the edge weight assignment constants, we choose $\alpha = 3$, $\beta = 1$ and $\gamma = 2$ that are the same as those in [8] for the trade-off between the RV insertion rate and the on-track RV insertion rate.

To investigate the timing impact of inserted redundant vias, we calculate the net sink node timing difference ratios after performing the conventional (without considering timing constraints) redundant via insertion method on each test circuit [8]. TABLE IV shows the difference ratio interval for each circuit. The ‘Timing Difference Ratio Interval’ is the timing disturbance ratio interval of net sink node delay differences after finishing the redundant via insertion procedure. It can be found that the range of timing difference ratio can be 35% ∼ 58%. Fig. 12 draws the sink node timing difference ratio probability graph for Primary1, and it reveals that the timing difference ratio interval can be very wide with non-ignorable probabilities. Therefore, it is meaningful to consider the timing effect while inserting redundant vias.

The results of redundant via insertion are presented in TABLE V. The “#Single Vias” is the total number of single via in the given design, “#Alive Vias” is the total number of alive vias that a alive via is a single via having at least one RV candidate, “#Ins. RVs” is the number of RVs inserted by executing the insertion algorithms, and “#Ins. Rate” is the RV insertion rate. The “#Ins. RVs w/o vio.” is equal to the total insertion number of RVs minus the number of inserted RVs causing timing constraint.

3The high ratio of positive timing difference is because the nets are short in test circuit [8]; therefore, the insertion of redundant via may have high effect on timing delay.
violation, “#Ins. Rate w/o vio.” is the RV insertion rate without violating timing constraints, and “Runtime of HMWM” is the runtime of HMWM algorithm [8]. The “Runtime of t-MWM with Increment.” is the runtime of the proposed t-MWM algorithm with utilizing the developed incremental timing analysis mechanism to update the required timing information, and “Runtime of t-MWM with non-Increment.” is the runtime of t-MWM algorithm with utilizing the conventionally non-incremental delay analysis method to update the timing information.

From TABLE V, the insertion rate can achieve to 99.68% in average without considering timing constraints. However, to meet the timing constraint, those inserted RVs located on the timing violation nets need to be removed. After removing those RVs, the insertion rate declines extremely to 90.20% in average. On the other hand, the insertion rate of the proposed RV insertion algorithm under timing constraints can achieve to 93.34% in average and guarantee the result satisfying timing constraints.

The last two columns in TABLE V also demonstrate that the developed incremental timing analysis technique can dramatically save the runtime of RV insertion procedure under timing constraints. Compared with the utilization of non-incremental timing analysis, it can speed up the runtime by over 34.398 times in average.

VII. CONCLUSION AND FUTURE WORK

In this paper, the RV insertion problem under timing constraints has been transformed to a MBC graph matching problem, and a developed two phase t-MWM algorithm utilizing the proposed incremental timing analysis method has been successfully used to solve the matching problem. The experimental results have showed that the insertion rate is improved, and how to fast and accurately estimate the timing behavior is crucial for the RV insertion under timing constraints.

Generally, some nets may not be the critical nets; hence, their timing delays after inserting redundant vias can be allowed to be worser than their original delays. To obtain the better and reasonable insertion rate, we are going to take the gate delay into consideration to fulfill the entire timing constraints in the future.

ACKNOWLEDGMENT

This work was supported in part by the National Science Council of Taiwan under Grant NSC 99-2220-E-009 -035.

REFERENCES